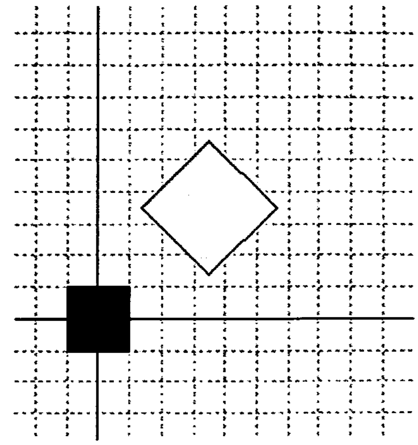
Exhibit F 09/491,864

Geometric transformation matrices

by Philippe de Lagrange

[Equations] [Matrix notation] [Combination example] [Conclusion]



Geometric transformations defined by so-called transformation matrices are linear applications from 3D space to 3D space. This kind of transformation can be a combination of:

- rotation
- scale (anisotropic)
- translation

A simple 2D example is shown on the figure on the right:
The red square is rotated 45 degrees, scaled by a factor sqrt(2), and translated to (3.5,3.5)

Transformation as equations

The transformation of the red square described above can be put in equations:

Let us call (x,y) the original coordinates and (x',y') the transformed coordinates.

$$x' = x - y + 3.5$$
$$y' = x + y + 3.5$$

Transformation as matrix

The equations can be rewritten as follows:

$$x' = a1.x + a2.y + t1$$
$$y' = a3.x + a4.y + t2$$

With proper values for *a1,a2,a3,a4,t1,t2*.
This is exactly the same as writing

X' = A . X + T

where $X'=\begin{matrix} x' \\ y' \end{matrix}$ $A=\begin{matrix} a1 & a2 \\ a3 & a4 \end{matrix}$ $X=\begin{matrix} x \\ y \end{matrix}$ $T=\begin{matrix} t1 \\ t2 \end{matrix}$

We have written the transformation in a matrix form, but we have two elements: A (rotation/scale) and T (translation).
The goal is to have only one transformation matrix. The annoyance comes from the translation (constant) term, (t1,t2).

The trick is to append a '1' at the end of X vector, and to append T on the right of A:

X' = B . X

x

where $X'=\begin{matrix}x'\\y'\end{matrix}$, $B=\begin{matrix}a1 & a2 & t1\\a3 & a4 & t2\end{matrix}$ $X=\begin{matrix}y\\1\end{matrix}$

This way, we still have the same set of equations, and we now have only one transformation matrix. But notation is not coherent any more: X and X' are not written the same way, and since B is not square, transformations can not be combined via matrix multiplication.

The final trick is to add the row [0 0 1] at the bottom of B, and to append a '1' at the end of X'. Then the transformation can be written as follows:

X' = M . X

where $X'=\begin{matrix}x'\\y'\\1\end{matrix}$ $M=\begin{matrix}a1 & a2 & t1\\a3 & a4 & t2\\0 & 0 & 1\end{matrix}$ $X=\begin{matrix}x\\y\\1\end{matrix}$

Now, we have one more equation: *1 = 1*, which is always true and perfectly useless, but allows us to have a coherent notation: transformation matrices are square and can be combined via matrix multiplication, and vector notation is consistent.

The goal has been reached at the cost of an additive row in each vector and transformation matrix.

Example of transformation combination

As said above, the transformation taken as an example is made of
- a sqrt(2) scale factor and a 45 degrees rotation. Let us call M1 this transformation.
- then, a (3.5,3.5) translation. Let us call it M2

$M1=\begin{matrix}1 & -1 & 0\\1 & 1 & 0\\0 & 0 & 1\end{matrix}$ $M2=\begin{matrix}1 & 0 & 3.5\\0 & 1 & 3.5\\0 & 0 & 1\end{matrix}$

The combined transformation M = M2 . M1 (beware of the order)

$M=\begin{matrix}1 & -1 & 3.5\\1 & 1 & 3.5\\0 & 0 & 1\end{matrix}$

Conclusion

In the 3D space, the mechanism is the same: vectors contain 4 elements (x,y,z,1), and matrices 4x4 (3x3 rotation/scale submatrix).

The matrix notation for geometric transformation is very smart and powerful from a theoretical point of view. But speaking of practical implementation, the last row/equation, which only denotes a notation trick, is completely useless. It is possible to get the same results without storing the last [0 0 0 1] row in matrices nor the last [1] element in vectors.

It can be noted the a transformation matrix is mainly made of
- a rotation/scale submatrix, which is the "same" size as vectors (3x3 for 3D vectors) and contain x,y,z coefficients in the associated equations,
- and a translation vector, which contains the constant terms in the associated equations.